

Prediction from Samples without Features

Ben Recht

October 10, 2025

In the last class, we discussed how optimal predictions at the population level can be computed by estimating the conditional probability of the label when a particular value of the feature vector is revealed. For today, let's take an unrealistic perspective and assume that we want to estimate the rate at which y is true when the feature is equal to x without using any of the other features. This very basic prediction problem will inform how we think about more complicated problems where we use relationships between different values of the features. It will help us set up the various frameworks people use when studying machine learning from samples.

In our simplified setting, we can throw the feature vector away and just think about the labels. We want to see how many samples are needed to build a good prediction for the remainder of the population. Let y_t be an arbitrary sequence of bits ($y_t \in \{0, 1\}$). We're going to explore different models to assess when the first T bits can yield a good predictor of the rest of the sequence.

Since it makes all of the calculations nice, we'll judge our predictors in the squared loss (Brier Score). Let m_k denote the average of the first k elements of the sequence $\{y_t\}$. That is $m_0 = 0$ and

$$m_k = \frac{1}{k} \sum_{i=1}^k y_i.$$

How might we evaluate prediction quality? Since we know we just want to estimate the mean of the full sequence, we could look at how well this mean predicts the mean of the sequence:

$$(m_T - m_N)^2$$

This is called *estimation error*. It is how well we can estimate a parameter given a finite set of data.

As another possibility, we might just be interested in how well it predicts the remaining bits on average:

$$\frac{1}{N-T} \sum_{t=1}^{N-T} (m_T - y_{T+t})^2.$$

This is called *prediction error*. Note that prediction is easier than estimation: the best predictor of the bits is the mean of the full sequence, so if you have a good estimate of the mean, you have a good prediction function. But, as we'll see, it is possible to have a poor estimate of the mean while still having near-optimal predictions.

The final model of prediction is *sequential*. In this case, we are interested in making predictions in an *online* setting, where we can adjust our predictor with each bit. The standard way to judge sequential predictions is called *regret*:

$$\frac{1}{T} \sum_{t=1}^{T-1} \left\{ (m_t - y_{t+1})^2 - (m_T - y_{t+1})^2 \right\}$$

In the first term in the summand, we are looking at how well the mean of the first t bits predicts bit $t + 1$. In the second term, we look at how well the average of all the bits predicts bit $t + 1$. Regret measures how quickly a sequential prediction algorithm mimics the best population prediction. We'd like the regret to rapidly converge to zero as a function of T .

1 iid Signals

First suppose the population of y_t are i.i.d. samples from a Bernoulli distribution with mean p . To simplify matters, in all of the analyses that follow, we'll only analyze the prediction error on the bit y_{T+1} . This will suffice to infer the prediction error on subsequent bits. We have

$$\mathbb{E}[(m_T - y_{T+1})^2] = \frac{T+1}{T} p(1-p).$$

To see this, just note

$$\begin{aligned} \mathbb{E}[(m_T - y_{T+1})^2] &= \mathbb{E}[(m_T - p) - (y_{T+1} - p)]^2 \\ &= p(1-p)/T + p(1-p). \end{aligned}$$

Our interpretation of this expectation is that the more samples you have, the better the prediction estimate becomes. However, note that even if T is very large, the prediction error is never better than $p(1-p)$. This is the variance of the process that generates each bit, and you can't expect to be able to predict with squared error better than this. If we are just predicting the individual bits, one sample is already enough to be within a factor of 2 of prediction via an infinite number of samples!

Note that by contrast

$$\mathbb{E}[(m_T - p)^2] = \frac{1}{T} p(1-p).$$

The estimate of the mean has a variance that goes to zero as T goes to infinity.

2 Without replacement samples

What happens when we sample T bits from the population uniformly at random? Note that this is equivalent to randomly ordering the data. In this case, let $\hat{p} = \frac{1}{N} \sum_{i=1}^N y_i$. If you work through the calculation, you will find

$$\mathbb{E}[(m_T - y_{T+1})^2] = \frac{N}{N-1} \cdot \frac{T+1}{T} \hat{p}(1-\hat{p}).$$

For large N , this is effectively the same prediction error as we calculated for the iid case.

3 Regret for Arbitrary Bit Sequences

The regret bound here parallels the standard analysis of strongly convex online gradient descent (See Hazan et al. [2006]). It's considerably more elegant for this particular bit estimation problem.

Theorem 3.1. *For any sequence $y_1, \dots, y_T \in \{0, 1\}$ we have*

$$\frac{1}{T} \sum_{t=1}^{T-1} (m_t - y_{t+1})^2 - \frac{1}{T} \sum_{t=1}^T (m_T - y_t)^2 \leq \frac{\log(T)}{T}.$$

This theorem says that, on average, the mean of the first t bits predicts the $t + 1$ bit about as well as the sample variance of the bits. It is a deterministic statement that holds for any sequence.

The proof of this statement follows from a structural lemma that equates the sample variance with a weighted sum of next token prediction errors. This equality shows that with appropriate weighting, the weighted squared error of the next-token predictor is equal to the mean-squared error of the sample mean.

Lemma 3.2. *For any sequence $y_t \in \mathbb{R}$ and any $z \in \mathbb{R}$,*

$$\sum_{k=0}^{T-1} \left(\frac{k}{k+1} \right) (m_k - y_{k+1})^2 - \sum_{k=0}^{T-1} (z - y_{k+1})^2 = -T(m_T - z)^2. \quad (1)$$

Proof [of Lemma 3.2] The Lemma is an immediate consequence of Welford's Identity [Welford, 1962]:

$$\sum_{k=0}^{T-1} (m_T - y_{k+1})^2 = \sum_{k=0}^{T-1} \left(\frac{k}{k+1} \right) (m_k - y_{k+1})^2 \quad (2)$$

A proof is included in the Appendix. To complete the proof of the lemma, note that for any z ,

$$\sum_{k=0}^{T-1} (z - y_{k+1})^2 = \sum_{k=0}^{T-1} (z - m_T + m_T - y_{k+1})^2 = \sum_{k=0}^{T-1} (m_T - y_{k+1})^2 + T(m_T - z)^2. \quad (3)$$

Replacing the first term on the right-hand side completes the proof.¹ ■

Theorem 3.1 now follows by applying two inequalities.

Proof [of Theorem 3.1] Since $y_k \in \{0, 1\}$ we can bound $|m_k - y_{k+1}| \leq 1$. Moreover, we have the following standard bound on the harmonic numbers

$$\sum_{k=1}^n \frac{1}{k} \leq 1 + \int_1^n \frac{1}{x} dx = 1 + \ln n$$

Putting these two upper bounds together proves the following formula for any z

$$\sum_{k=0}^{T-1} (m_k - y_{k+1})^2 - \sum_{k=0}^{T-1} (z - y_{k+1})^2 \leq -(z - m_T)^2 + \log(T) + 1.$$

¹Thanks to Ryan McCorvie for pointing me to this connection with Welford's identity.

Letting $z = m_T$ proves the theorem. ■

Note again that $\sum_{k=0}^{T-1} (z - y_{k+1})^2$ is the sample variance of the first T elements of the sequence. It shows, roughly, that using the mean to predict the future is close to the best we can hope to do in mean-squared error, even with no probabilistic assumptions about the data.

References

Elad Hazan, Adam Kalai, Satyen Kale, and Amit Agarwal. Logarithmic regret algorithms for online convex optimization. In *COLT*, 2006.

B. P. Welford. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420, 1962.

A Welford’s Identity

This derivation is directly copied from Welford’s paper, but I’m including it here for those who don’t have access to JSTOR:

$$\begin{aligned} \sum_{k=0}^{T-1} (y_{k+1} - m_T)^2 &= \sum_{k=0}^{T-2} \left[(y_{k+1} - m_{T-1}) - \frac{1}{T}(y_T - m_{T-1}) \right]^2 + \left(\frac{T-1}{T} \right) (y_T - m_T)^2 \\ &= \sum_{k=0}^{T-2} (y_{k+1} - m_{T-1})^2 + \left[\frac{T-1}{T} + \frac{(T-1)^2}{T^2} \right] (y_T - m_T)^2 \\ &= \sum_{k=0}^{T-2} (y_{k+1} - m_{T-1})^2 + \frac{T-1}{T} (y_T - m_T)^2. \end{aligned}$$

The identity (2) follows by unwrapping this recursion.